Principal Components Analysis and Singular Value Decomposition

Principal Components Analysis (PCA)

Principal Component Analysis, or PCA, is a well-known and widely used technique applicable to a wide variety of applications such as dimensionality reduction, data compression, feature extraction, and visualization. The basic idea is to project a dataset from many correlated coordinates onto fewer uncorrelated coordinates called principal components while still retaining most of the variability present in the data.

Principal Component Analysis (PCA) is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

So, to sum up, the idea of PCA is simple, it reduces the number of variables of a data set, while preserving as much information as possible.

Summary:

- Principal Component Analysis (PCA) is a dimensionality reduction method.
- The goal is to embed data in high dimensional space, onto a small number of dimensions.
- It most frequent use is in exploratory data analysis and visualization.
- It can also be helpful in regression (linear or logistic) where we can transform input variables into a smaller number of predictors for modeling.

EXAMPLE - PCA

A

$$X = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix}$$

From spectral theorem:

$$(X^T X)\phi = \lambda\phi$$

$$X^T X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix}$$

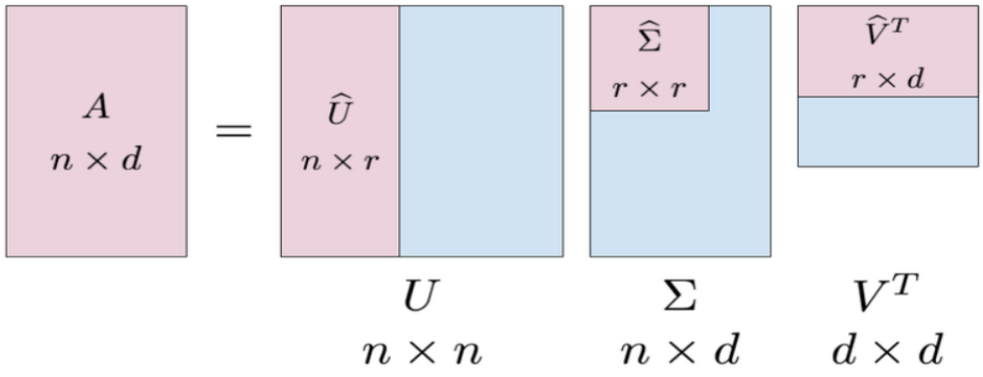
B

$$X = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix} \qquad \phi_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \qquad \lambda_1 = 58 \qquad \phi_2 = \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \qquad \lambda_2 = 2$$

Singular Value Decomposition (SVD)

Singular Value Decomposition, or SVD, is a computational method often employed to calculate principal components for a dataset. Using SVD to perform PCA is efficient and numerically robust. Moreover, the intimate relationship between them can guide our intuition about what PCA actually does and help us gain additional insights into this technique.

Singular Value Decomposition is a matrix factorization method utilized in many numerical applications of linear algebra such as PCA. This technique enhances our understanding of what principal components are and provides a robust computational framework that lets us compute them accurately for more datasets.



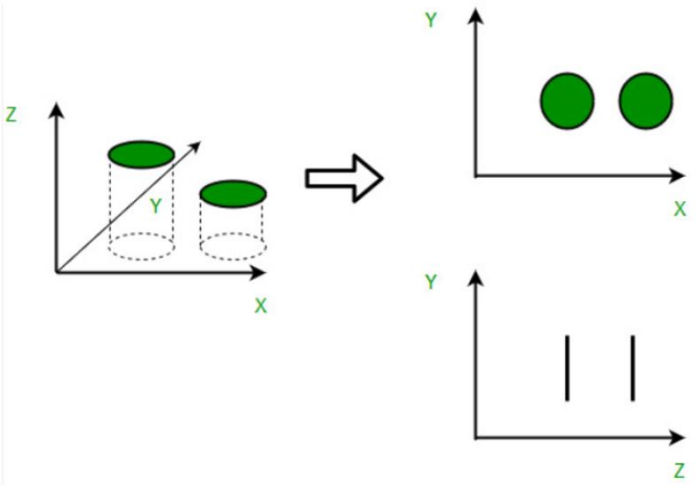
Why dimensionality reduction?

Dimensionality reduction

In machine learning, “dimensionality” simply refers to the number of features (i.e. input variables) in your dataset.

While the performance of any machine learning model increases if we add additional features/dimensions, at some point a further insertion leads to performance degradation that is when the number of features is very large commensurate with the number of observations in your dataset, several linear algorithms strive hard to train efficient models. This is called the “Curse of Dimensionality”.

Dimensionality reduction is a set of techniques that studies how to shrivel the size of data while preserving the most important information and further eliminating the curse of dimensionality. It plays an important role in the performance of classification and clustering problems.



Dimensionality reduction, or **dimension reduction**, helps to transform data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse as a consequence of the curse of dimensionality, and analyzing the data is usually computationally intractable. Dimensionality reduction is common in fields that deal with large numbers of observations and/or large numbers of variables, such as signal processing, speech recognition, neuroinformatic, and bioinformatics.

Techniques for dimensionality reduction

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Generalized Discriminant Analysis (GDA)
- Multi-Dimension Scaling (MDS)
- LLE
- IsoMap
- Autoencoders

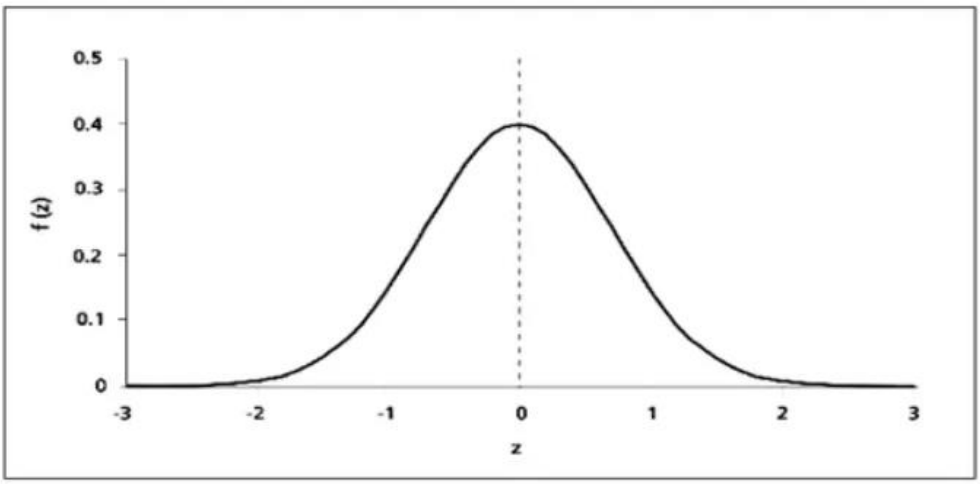
Z-score standardization

What is Z-score?

A z-score, or standard score, is used for standardizing scores on the same scale by dividing a score's deviation by the standard deviation in a data set. The result is a standard score. It measures the number of standard deviations that a given data point is from the mean.

A z-score can be negative or positive. A negative score indicates a value less than the mean, and a positive score indicates a value greater than the mean. The average of every z-score for a data set is zero.

A z-score describes the position of a raw score in terms of its distance from the mean, when measured in standard deviation units. The z-score is positive if the value lies above the mean, and negative if it lies below the mean. It is also known as a standard score, because it allows comparison of scores on different kinds of variables by standardizing the distribution. A standard normal distribution (SND) is a normally shaped distribution with a mean of 0 and a standard deviation (SD) of 1.



A standard normal distribution (SND)

Why are z-scores important?

It is useful to standardized the values (raw scores) of a normal distribution by converting them into z-scores because:

(a) it allows researchers to calculate the probability of a score occurring within a standard normal distribution;

(b) and enables us to compare two scores that are from different samples (which may have different means and standard deviations).

How do you interpret a z-score?

The value of the z-score tells you how many standard deviations you are away from the mean. If a z-score is equal to 0, it is on the mean.

A positive z-score indicates the raw score is higher than the mean average. For example, if a z-score is equal to +1, it is 1 standard deviation above the mean.

A negative z-score reveals the raw score is below the mean average. For example, if a z-score is equal to -2, it is 2 standard deviations below the mean.

Z-Score Formula

= STANDARDIZE (x, mean, standard_dev)

The STANDARDIZE function uses the following arguments:

X (required argument) – This is the value that we want to normalize.

Mean (required argument) – The arithmetic mean of the distribution.

Standard_dev (required argument) – This is the standard deviation of the distribution.